ЗАЩИЩЕННЫЙ ОБМЕН ДАННЫМИ ПО ПРОТОКОЛУ PEER-TO-PEER

Шитько А. М.

УО «Белорусский государственный технологический университет», Минск, Беларусь, e-mail: a.shitko@belstu.by

Одной из разновидностью peer-to-peer сети является частично децентрализованная (гибридная), в которой существует сервер, используемый для координации, поиска или предоставления информации о существующих участниках сети и их статусе.

Сервер спроектирован на языке Java с использованием Spring Framework [1]. В качестве реализации p2p-протокола используется библиотека JXTA [2]. На сервере установлен SSL-сертификат для организации защищенного соединения с участниками сети. В качестве хостинга настроена Google App Engine. Вся передаваемая пользователями информация храниться в Google Cloud SQL, которая представляет собой реляционную базу данных, хранящаяся в «облаке» Google. Данные удаляются после окончания сеанса работы.

Важным преимуществом системы является упрощенная (для пользователя) регистрация на основе протокола авторизации OAuth 2.0. Он позволяет выдавать права на доступ к ресурсам пользователя без предоставления его логина и пароля. Вместо этого каждый пользователь получает «токен» доступа, который является результатом авторизации и пропуском к защищенному ресурсу.

В ходе процедуры регистрации «пир» посылает get-запрос на OAuth-сервер, в теле которого содержатся идентификатор клиента, полученный сервером социальной сети, URL перенаправления, на который будет направлен пользователь после успешной авторизации, и тип ответа «code» для получения кода от сервера. Дополнительно для пользователя создается универсальный уникальный идентификатор (UUID), генерируемый средствами библиотеки JXTA при создании объекта «Peer» и отсылается на сервер. После успешной авторизации на Spring-сервер отсылается «code». Далее для получения токена доступа с сервера посылается get-запрос, в теле которого находятся идентификатор клиента, секретный ключ, предварительно полученный от сервера социальной сети, «code» и URL перенаправления. После на сервер приходит ответ в формате JSON с токеном доступа и идентификатором пользователя. Для получения персональных данных пользователя необходимо сделать еще один get-запрос на объект «users.get», вследствие чего возвращается ответ в формате JSON с персональными данными клиента. Все полученные данные сохраняются в базе данных на сервере.

В качестве «пиров» выступают мобильные устройства на платформе Android [3]. Поиск участников в сети и передача им некоторых данных осуществляется по соответствующим идентификаторам UUID, генерируемые для каждого пользователя. Обмен данными происходит по зашифрованным двунаправленным именованным каналам.

Литература

1. Хо, К. Spring для профессионалов / К. Хо, Р. Харроп. – Москва: Вильямс, 2012. – 880 с.
1. Practical JXTA II [Электронный ресурс] / Scribd. – 2011. – Режим доступа: http://www.scribd.com/doc/47538921/Practical-JXTA-II. – Дата доступа: 17.09.2014.